




Automatic Tenement Identification — Case Study

SoftTech Engineers Limited | [2025] – Present

Problem Statement

In architectural and construction workflows, creating and managing tenements (groups of spatial units such as residential or commercial spaces) within large-scale 3D building models is a critical but time-consuming task. Traditionally, architects and modelers manually group rooms or spaces into tenements and assign names or classifications. This manual process is:

-  **Highly labor-intensive**, often consuming up to **60%** of the model preparation time.
-  **Prone to human error and inconsistency.**
-  **Difficult to scale** for buildings with hundreds of floors and thousands of spaces.



Given the increasing complexity and size of building information models, there was a clear need to automate tenement identification to improve **efficiency**, **accuracy**, and **scalability**.

Existing System

The existing system relied heavily on manual intervention:

- Architects would place individual rooms or spaces within the 3D model.
- They manually created tenements by grouping these spaces based on their judgment and project requirements.
- Naming and organizing these groups was also performed manually.



This approach:

-  Resulted in significant time consumption without automated verification or optimization.
-  Lacked scalability and was inefficient for high-density building projects.

Proposed Solution

We designed and developed an **automated backend service** to identify and form tenement groups directly from spatial data within the building model, eliminating the need for manual grouping.

Key insight: Model the problem as a **graph connectivity challenge**, where:

-  **Nodes** represent individual spatial units (rooms, spaces).
-  **Edges** represent connections between spaces via architectural elements such as doors or openings, indicating accessibility.

By applying graph traversal algorithms to this model, the system automatically detects clusters of connected spaces that constitute a tenement.

Technical Approach

Input Processing

- The service accepts a 3D building model file as input and extracts spatial units and their connectivity data based on architectural elements (doors, openings).

Graph Construction

- Each space is represented as a node, and edges are created for each identified connection between spaces.

Connected Components Detection

- The algorithm performs a **breadth-first search (BFS)** on the graph to identify connected components, where each connected component corresponds to a tenement.

Tenement Formation

- Using domain-specific rules (e.g., room types like residential or commercial), the service groups spaces within each connected component into tenements.

Output Generation

- The system outputs an updated model with tenements created and an accompanying JSON file detailing the tenement groups and their constituent spaces.

Benefits and Impact

Metric	Before	After
Manual Grouping Effort	High (~60% of model prep time)	Reduced by approximately 40%
Accuracy	Subject to human error	Deterministic and consistent results
Scalability	Challenging for complex models	Fully automated and scalable solution
Time Efficiency	Limited by manual processes	Significantly improved workflow speed

This automation dramatically reduces manual effort, improves accuracy, and enables architects and engineers to focus on higher-value tasks.

Technologies Used

- **C# and .NET** for backend service development
- **Microservices architecture** for modularity and scalability
- **Graph algorithms (BFS)** for spatial connectivity analysis

Summary

By reframing the problem of tenement identification as a **graph connectivity challenge** and applying well-established graph algorithms, we successfully automated a previously manual, error-prone task. This solution delivers significant **time savings** and improved **scalability** in managing large and complex spatial building models.

Curious to learn more or see a demo?

Feel free to ask about the technical details, code snippets, or integration strategies!